Doc Code: AP.PRE.REQ

PTO/SB/33 (01-09)
Approved for use through 02/28/2009. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PRE-APPEAL BRIEF REQUEST FOR REVIEW

Docket Number (Optional)
NetSuite 709 (formerly NetLedger 701)

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to "Mail Stop AF, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450" [37 CFR 1.8(a)]

on March 30, 2009

Signature Robert Moll

Typed or printed name Robert Moll

Application Number	Filed
10/071,496	March 1, 2002

First Named Inventor: Brian Chess

Art Unit	Examiner
2157	Avi M. Gold

Applicant requests review of the final rejection in the above-identified application. No amendments are being filed with this request.

This request is being filed with a notice of appeal.

The review is requested for the reason(s) stated on the attached sheet(s).
Note: No more than five (5) pages may be provided.

I am the

- ☐ applicant/inventor.
- ☐ assignee of record of the entire interest. See 37 CFR 3.71. Statement under 37 CFR 3.73(b) is enclosed. (Form PTO/SB/96)
- ☑ attorney or agent of record. Registration number 33,741
- ☐ attorney or agent acting under 37 CFR 1.34. Registration number if acting under 37 CFR 1.34 _____

Robert Moll
Signature

Robert Moll
Typed or printed name

650-567-9153
Telephone number

March 30, 2009
Date

NOTE: Signatures of all the inventors or assignees of record of the entire interest or their representative(s) are required. Submit multiple forms if more than one signature is required, see below*.

☑ *Total of One forms are submitted.

PRE-APPEAL BRIEF REQUEST FOR REVIEW ARGUMENTS

The invention relates to client-side caching systems and methods that reduce page latency and bandwidth usage of Internet based and web applications.

In an embodiment, a server computes a cookie value based on the last version of a resource (e.g. a web page). When an application user uses a browser to request a page from the server, the server responds with a small non-displayed page and a client-side script. The client-side script appends the cookie value to the original request and the browser automatically re-requests the page with the appended cookie value. If the most recent version of the page is in the browser cache, the browser gets a cache hit, which means the page is retrieved from the browser cache rather than from the server, rapidly displaying the page to the user. If, on the other hand, there is only an older version of the page in the browser cache, there is a cache miss (because the client-side's rewritten request will not match any request in the browser cache) and the browser will send the request to the server to retrieve the most recent version of the page.

Claim 1 recites a client-side caching system, comprising:

a client for issuing a request based on user selection for a resource on a server; and

a server for sending a response including a cookie and a client-side script to the client, wherein the cookie value represents the last version of the resource, and the client-side script appends the cookie value to the request for a resource such that the client automatically re-requests the resource with the appended cookie value so that if the most recent version of the resource is in the client cache, the resource is retrieved from the client cache rather than from the server, and if not, the resource is retrieved from the server.

In sections 1-2, the Office action rejects claims 1-18 as being unpatentable over U.S. Patent No. 6,510,439 B1 to Rangarajan et al. (Rangarajan) and U.S. Patent Publication No. 2002/0032701 to Gao et al. (Gao).

Specifically, the Office action on pages 2-3 asserts (1) Rangarajan teaches a client-side caching system except a client-side script that automatically re-requests a resource, (2) Gao teaches a client-side script that automatically requests updated data, and (3) it would have been obvious to one of ordinary skill at the time of the invention to modify Rangarajan in view of Gao to use a client-side script that automatically re-requests a resource. One would be motivated to do so because it is more efficient for the script to run on the client.

Rangarajan and Gao cannot establish a prima facie case of obviousness, because they fail to disclose the client-side script function recited in claim 1:

1) Claim 1 requires a client-side script function that appends the cookie value to the request for a resource such that the client automatically re-requests the resource with the appended cookie value so that if the most recent version of the resource is in the client cache, the resource is retrieved from the client cache rather than from the server, and if not, the resource is retrieved from the server.

2) The Office action concedes that: "Rangajan fails to teach the limitation further including the use of client-side script that automatically re-requests a resource" (See page 3, paragraph 3).

3) However, on page 3, paragraph 4, the Office action asserts this missing limitation is disclosed by Gao (paragraph 0047).

4) However, Gao fails to disclose it (See Gao paragraphs 0045-0051). As shown in Figure 5, a client 102 sends a request for a web page to the web server 104 (entry "1"), the web server returns an HTML file to the client (entry "2"), the client displays the file and requests additional files (e.g. images)(entry "3"), the web server returns the requested files and generates a Javascript file that will cause the client's browsers to generate a request for update data (entry "4").

5) Gao states: "When the JavaScript code is received, it will be automatically interpreted by the browser, which will cause the script program to

be executed by the browser. In accordance with the invention, the JavaScript code will cause the browser to request another page, for a second Web page. This operation is represented by the fifth entry of FIG. 5, "5. RECEIVE SCRIPT, INTERPRET CODE, REQUEST UPDATE PAGE." The JavaScript code in accordance with the invention specifies that the requested second page will not be displayed by the browser. That is, the data contained in the second (update) page will be received, but no page will be displayed. Thus, the requested page comprises a "phantom" page (hence the dashed line representation of FIG. 2). When the Web server receives the request for the phantom page after operation (5), it will generate the requested data and will send it (otherwise formatted as an HTML page) to the browser. The sending operation is represented in FIG. 5 by the sixth entry, "6. SEND REQUESTED HTML DATA FOR PHANTOM PAGE" (Gao paragraph 0049, emphasis added).

6) Gao's JavaScript code causing the browser to request a second Web page indicates it is not a "re-request of the resource" that is, the first page requested by the user. Further, no user requests a non-displayed second page, known as a phantom page (Gao paragraph 0049). And what happened to the client-side script that appends a cookie value to the request in Gao? Gao discloses no cookie much less a cookie representing the last version of the first web page. The Office action limits its attention as to whether the Gao client-side script causes the client to automatically re-request the resource and it doesn't.

7) Gao's paragraph 0050 says the client browser receives the requested update data, then the browser processes it according to the client-side script for display in the first web page. Could this update data be "the resource" first requested by the user? No, that is the first web page. This update data is not being "re-requested." This update data is being requested for the first time! As shown in Figure 5 and paragraphs 0046-0050, Gao requests the first web page once (entry "1"), requests the update data (update page) once (entry "5") then displays them together (entry "7"). Thus, Gao fails to disclose a client-side script such that the client "automatically re-requests the resource" as recited in claim 1.

8) It is legal error to ignore these claim limitations in analysis. See e.g., *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988) where the Federal Circuit held a reference did not render a claim prima facie obvious because *the examiner ignored that a claim limitation was absent in the reference.*

9) In the amendment dated July 7, 2008, the applicants argued that Rangarajan does not disclose a client-side caching system because Ragarajan never uses the term "cache" or "caching" or "client cache" or "client-side cache." The Office action disagreed. On page 11, the Office action states "documents are kept/cached on the client" and on Rangarajan col. 7, lines 8-44, "there is an appropriate document sent to a client via a URL, which would be inherently stored in ... the client cache." In response to Office action on pages 11-12, we incorporate by reference the remarks of our amendment of July 7, 2008. Further, whether or not Rangarajan and Gao describe a client-side caching system misses the point: they fail to disclose the client-script function recited in claim 1 that solves the problem of reliably providing the last version of the resource while reducing latency and bandwidth usage.

In view of the above, claim 1 is allowable over Rangarajan and Gao.

Claims 2-4 are allowable due to dependency on claim 1.

Contrary to the Office action on page 4, Rangarajan fails to describe the response includes a non-displayed relatively small page and fails to describe a client-side script in the entity body of the response as recited in claim 3.

Contrary to the Office action on pages 4-5, Rangarajan does not teach claim 5 for the reasons discussed in connection with claim 1.

Contrary to the Office action on page 5, Rangarajan does not teach an application server inserting a client-side script into the entity body of the response as recited in claim 6. In fact, Rangarajan fails to even describe "a server sending a script to the client" as asserted on page 3 of the Office action. Rangarajan's Figure 1 and

col. 7, lines 8-16 and col. 7, lines 31-44 describe the server-side (e.g., a CGI program, state management server 12, registration table 12, HTTP server 16) in a dotted box and the client-side as end user 24. Rangarajan's col. 9, line 65 - col. 10, line 11 describe the HTTP server as sending a cookie and a document to a client.

Claim 7 is separately patentable, since Rangarajan fails to describe the server setting the cookie value by determining the last modified time of each web page in the same class as the web page which is the subject of the request, and setting the cookie value to the maximum value of the last modified times.

Claim 8 is separately patentable, since Rangarajan fails to describe the server setting the cookie value by determining the last modified time of each web page in the same class as the web page which is the subject of the request, and setting the cookie value to the maximum value of the last modified times.

Contrary to the Office action on pages 6-8, Rangarajan and Gao do not teach the claims 9-14 for reasons similar to that discussed in connection with claim 1.

Claim 14 is separately patentable, since Rangarajan fails to describe the server setting the cookie value by determining the last modified time of each page in the same class as the page which is the subject of the request, and setting the cookie value to the maximum value of the last modified times.

Contrary to the Office action on pages 8-9, Rangarajan and Gao do not teach claims 15-17 as discussed in connection with claim 1.

Contrary to the Office action on page 10, Rangarajan and Gao do not teach claim 18 for reasons similar to that discussed in connection with claim 1. It is respectfully submitted that the application is in condition for allowance.

Respectfully Submitted,

Robert Moll

Robert Moll, Reg. No. 33,741